



Trabalho Prático 3

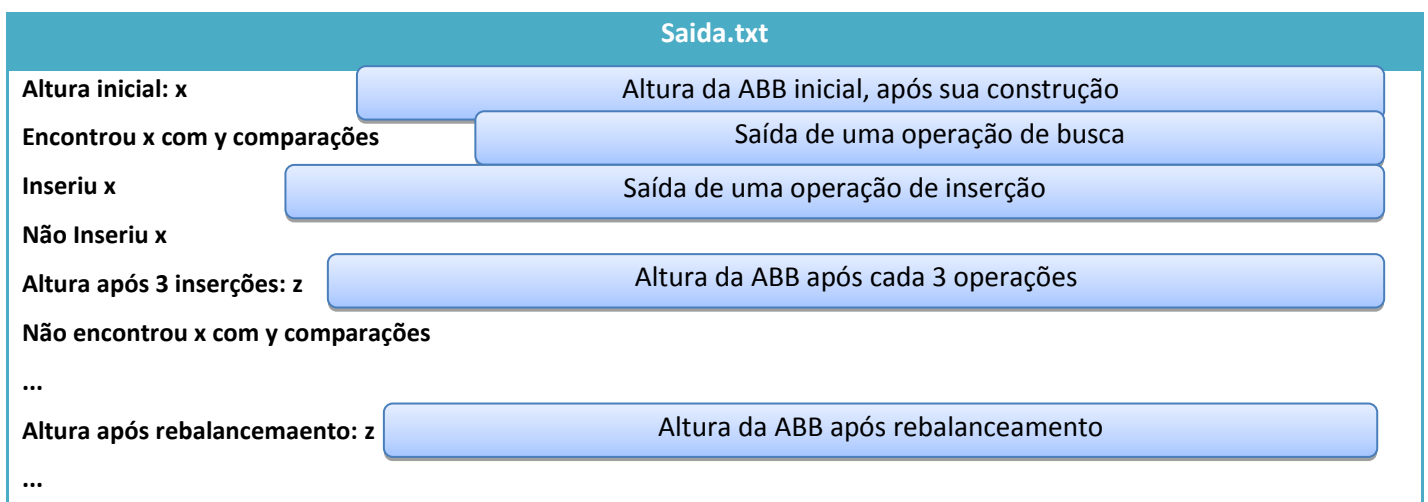
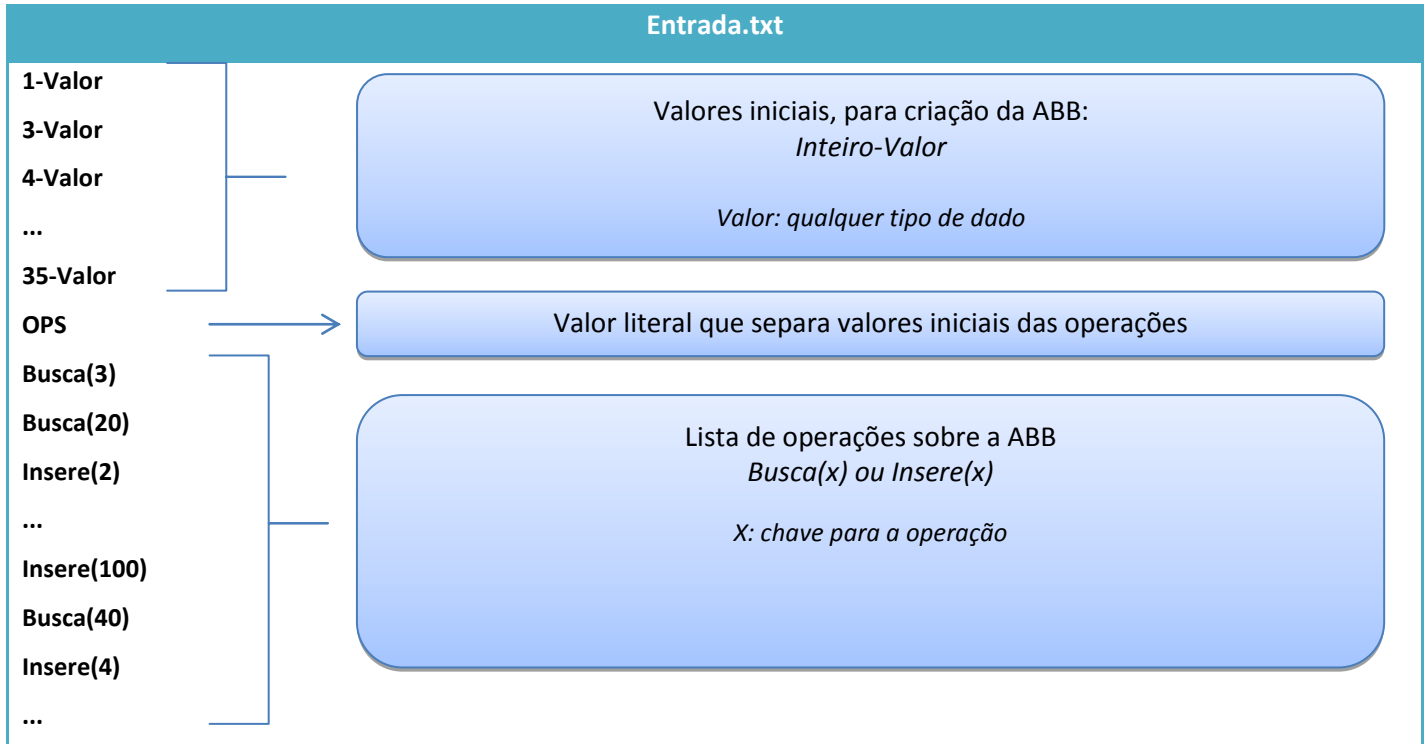
Data de divulgação: 17/11/10
Prazo para entrega: 03/12/10

Descrição do Problema

Defina um TAD Árvore de Busca Binária (ABB). Implemente uma aplicação com as seguintes características:

- a) Sejam n registros organizados como uma ABB em relação a um campo chave de tipo inteiro. O registro deve ser composto da chave e de outro valor (de tipo arbitrário, por exemplo, *char* ou *string*).
- b) A partir de um arquivo de entrada (e não a partir do console), com os registros ordenados em ordem crescente das chaves, construa inicialmente a ABB de forma que ela tenha a altura mínima possível. Feita a criação, calcule a altura da árvore e coloque num arquivo de saída.
- c) Após construída a ABB, efetue várias operações de **busca** e **inserção** de registros. Essas operações devem estar especificadas no arquivo de entrada, seguindo os dados de criação da árvore (do item anterior). Os resultados devem ser apresentados num arquivo de saída. Para cada operação de busca, exibir o número de comparações entre chaves efetuadas no arquivo de saída; para cada inserção, se o registro foi inserido ou não.
- d) Calcule a altura da árvore no final de cada 3 operações de inserção.
- e) Após a execução de 10 operações de inserção, aplique o algoritmo de rebalanceamento da árvore (percorrendo-a em *in-ordem*). Após o balanceamento, calcular a altura da árvore e colocar o resultado no arquivo de saída.

Exemplo de arquivo de entrada e saída



Ferramentas

A implementação do trabalho será em linguagem C, utilizando o compilador GCC.



Diversos ambientes de programação utilizam o GCC como compilador padrão, com é o caso do Dev-C++ (<http://dev-c.softonic.com.br/>) e o Code Block (<http://www.codeblocks.org/>), inclusive esse compilador é o padrão da maioria das distribuição Linux.

Caso deseje instalar esse compilador para ser usado em linha de comando nos sistemas Windows, baixar o MinGW (<http://www.mingw.org/>, baixar em <http://sourceforge.net/projects/mingw/>).

Entrega do Trabalho

Deve ser entregue, até a data determinada, um arquivo zipado chamado GrupoXX.zip, onde XX corresponde ao número do seu grupo, contendo os arquivos abaixo:

- **ArvoreBB.c** (TAD da implementação da Árvore de Busca Binária (ABB));
- **ArvoreBB.h** (*Header* do TAD da implementação da ABB);
- **Principal.c** (*Header* do TAD da implementação da ABB);
- **Entrada.txt** (Arquivo de texto com os dados de entrada do programa);
- **Saida.txt** (Arquivo de texto com as saídas do programa);

A entrega do arquivo deve ser pelo email arquivos.maziero@gmail.com. Serão considerados os email enviados até às 23:59 do dia 03/12/2010. Trabalhos fora do prazo serão penalizados (como descrito abaixo).

Critérios de Avaliação e Penalidade

A nota do trabalho 1 (NT1) será constituída da seguinte maneira:

$$NT1 = 0.4*NP1 + 0.4*NP2 + 0.2*NP3, \text{ onde:}$$

NP1 é a nota do TAD Clientes (implementação usando lista sequencial);

NP2 é a nota do TAD Transações (implementação usando lista sequencial);

NP3 é a nota do programa principal.

As notas dos programas **NPx** terão os seguintes critérios, com os respectivos pesos:

$$NPx = 0.5*NC + 0.2*NE + 0.1*NI + 0.2*NCF, \text{ onde:}$$



NC é a nota do critério Correção;

NE é a nota do critério Eficiência;

NI é a nota do critério Interface;

NCF é a nota do critério Código Fonte.

• **Correção (NC):** o programa faz o que foi solicitado? Faz tudo o que foi solicitado?

Utiliza encapsulamento de informação? Erão considerados nos critérios de avaliação:

- O uso correto dos TAD e operações de manutenção relacionadas;
- Cumprimento dos requisitos funcionais, ou seja, realiza as transações da maneira correta;

• **Eficiência (NE):** as operações são executadas da maneira mais eficiente para cada estrutura de dados? Evita código duplicado/redundante/não atingível?

• **Interface (NI):** é simples de usar, prático, tolera os erros mais óbvios? O trabalho foi entregue dentro das especificações (zipado, com os nomes de arquivo solicitados)?

- Interface do programa;
- Interface de entrega do trabalho;

• **Código fonte (NCF):** é claro e organizado? Nomes de variáveis são sugestivos? Está bem documentado?

- Clareza;
- Nomes de variáveis;
- Documentação/comentários no código.

A nota do trabalho sofrerá penalidade se entregue depois do prazo estipulado, conforme os critérios abaixo:

NT1 = $k \cdot NT1$, onde k é um fator multiplicador dado por:

$k = 1.0$, se o trabalho for entregue dentro do prazo determinado (não há penalidade);

$k = 0.7$, se o trabalho for entregue até 24hs após o prazo determinado;

$k = 0.5$, se o trabalho for entregue entre 24hs e 48hs após o prazo estipulado;

$k = 0.0$, se o trabalho for entregue mais de 48hs após o prazo estipulado (será considerado desistente).

Para cada trabalho recebido por e-mail será enviada uma resposta de confirmação!